

A multi-constraint learning path recommendation algorithm based on knowledge map

Zhu, H, Tian, F, Wu, K, Shah, N, Chen, Y, Ni, Y, Zhang, X, Chao, K-M & Zheng, Q

Author post-print (accepted) deposited by Coventry University's Repository

Original citation & hyperlink:

Zhu, H, Tian, F, Wu, K, Shah, N, Chen, Y, Ni, Y, Zhang, X, Chao, K-M & Zheng, Q 2017, 'A multi-constraint learning path recommendation algorithm based on knowledge map' *Knowledge-Based Systems*, vol 143, 102-114pp.

<https://dx.doi.org/10.1016/j.knosys.2017.12.011>

DOI 10.1016/j.knosys.2017.12.011

ISSN 0950-7051

ESSN 1872-7409

Publisher: Elsevier

NOTICE: this is the author's version of a work that was accepted for publication in *Knowledge-Based Systems*. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in *Knowledge-Based Systems*, [143, (2017)] DOI: 10.1016/j.knosys.2017.12.011

© 2017, Elsevier. Licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International

<http://creativecommons.org/licenses/by-nc-nd/4.0/>

Copyright © and Moral Rights are retained by the author(s) and/ or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This item cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder(s). The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

This document is the author's post-print version, incorporating any revisions agreed during the peer-review process. Some differences between the published version and this version may remain and you are advised to consult the published version if you wish to cite from it.

A Multi-Constraint Learning Path Recommendation Algorithm Based on Knowledge Map

Haiping Zhu^{1,2}, Feng Tian^{2,3*}, Ke Wu^{1,2}, Nazaraf Shah⁴, Yan Chen^{1,2}, Yifu Ni^{1,2}, Xinhui Zhang¹,
Kuo-Ming Chao⁴, Qinghua Zheng^{1,2}

¹Department of Computer Science and Technology, Xi'an Jiaotong University

² MOEKLINNS Lab, Department of Computer Science and Technology, Xi'an Jiaotong
University, 710049, China

³Systems Engineering Institute, Xi'an Jiaotong University

⁴Department of Computing, Coventry University, CV12JH, UK

Abstract—It is difficult for e-learners to make decision on how to learn when they are facing with large amount of learning resources, especially when they need to balance available limited learning time and multiple learning objectives in various learning scenarios. This paper attempts to address this challenge by proposing a new multi-constraint learning path recommendation algorithm based on knowledge map. The main contributions of the paper are as follows. Firstly, two hypotheses on the e-learners' need of different learning paths for four different learning scenarios (initial learning, usual review, pre-exam learning and pre-exam review) are verified through questionnaire-based statistical analysis. Secondly, according to learning behavior characteristics of four types of the learning scenarios, seven kinds of learning path constraint factors are proposed to determine different kinds of learning paths, such as shortest learning path, critical learning path and easy learning path. Thirdly, the proposed multi-constraint learning path recommendation algorithm based on the knowledge map is implemented and it combines the domain knowledge structure and cognitive structure of the learners to meet their need on different learning paths for the different learning scenarios. Finally, the results calculated from questionnaires verifies the statistical similarity between the learners' self-organized learning path and the recommended learning path.

Keywords—E-learning; Knowledge map; Learning scenario; Learning path recommendation

1 Introduction

Constructing a learning path is a well researched topic in personalized recommendation [1]. In SCORM [2], a learning path represents a sequence of learning activities. Selecting proper learning objects (LOs) to compose a suitable learning path for e-learners is a complex task [3][4], especially considering available limited learning time and multiple learning objectives in various learning scenarios. For example, with the rapid development of mobile learning (m-

learning), M-learning is characterized both by fragmented learning time and fragmented learning resources for e-learners, which influences various learning paths for different learning scenarios in a learning procedure. This brings new challenges for learning path recommendation.

It is necessary to consider the diverse needs of e-learners in different learning scenarios. In the existing studies on scenario based-learning (SBL) **Error! Reference source not found.** and goal based scenario (GBS) [7], the main focus is on influence of scenario on users. However, this research strand lacks the consideration of the learner's different learning path preferences in different learning scenarios. According to the related research[8-11], we have realized that a learner has his own preferred learning path for a specific learning scenario. Correspondingly, many learners in the same learning scenarios may share one or two common kinds of learning paths, as can be seen in experiment results presented in Section 4.

There is a tendency that e-learners use time fragments to learn. For example, based on the analysis of 12600 students' learning log of 40 courses from the Distance Learning College (DLC) of our school, the results show that more than 80% of learning durations that each person studies continuously, is less than 3 minutes. Therefore, it is necessary to provide an appropriate learning path that directs e-learner's focuses on the interested knowledge units (also called knowledge elements) within limited time constraints.

The organization of learning resources also directly affects the learning path recommendation, while the learning time fragments put forward a need that a recommended learning path should better be composed of many fine-grained learning resources. Most research on learning resource recommendation methods have ignored the inherent dependency between human cognitive characteristics and knowledge units. Although a few of them have considered the inherent dependency, however they have neglected other key factors such as different scenarios, fine-grained learning resources and limited time.

To address these problems our paper presents a new multi-constraint learning path recommendation algorithm to meet the diverse needs of learners in different learning scenarios. The main contributions of this paper are as follow.

(1) Our paper presents four basic learning scenarios (initial learning, usual review, pre-exam learning and pre-exam review) and provides two hypotheses (Hypothesis I: a learner has different learning path requirements in different learning scenarios. Hypothesis II: different learners have similar learning path requirements in a same learning scenario). Note that a learning scenario here refers to a given type of learning situation which is closely related to

one user's learning process. After design distribution and collection of the questionnaires, we obtain the learners' self-organized learning paths in different scenarios and the learners' rating for the recommended learning paths. The analysis of the results based on the collected questionnaires proves these two hypotheses.

(2) To meet the learners' diverse needs in different learning scenarios in fragmented learning time, eight different learning paths are presented, including complemented learning path, shortest learning path, shortest-duration learning path, critical learning path, easy learning path, complete learning path, more-hotspot learning path and the quick learning path. The constraint factors of different types of paths are presented and included in the proposed multi-constraint learning path recommendation model to adjust adaptively.

(3) We propose a multi-constrained learning path recommendation algorithm based on knowledge map[12], as well as other key information including the learning logs and the features of learners' behaviors and learning resources, such as learning duration, learning frequency, learning interval, attention degree and learning centrality of knowledge unit(KU). The feature of learning centrality of KU is based on our team's previous research [13], in which the inheritance and development of knowledge is seen as the stochastic dynamic migration process of the semantic information in Knowledge Map.

The rest of the paper is organized as follows. Section 2 presents relevant research on the learning path recommendation. Section 3 provides definitions of knowledge units, knowledge map and learning path. Section 4 describes multi-constraint model of learning path and the implemented multi-constraint learning path recommendation algorithm. Section 5 presents experimental result and analysis. Assembly Language course is used as an example. Questionnaire and subjective evaluation are used for verifying hypothesis mentioned above and for the effectiveness of the recommended learning paths for different learning scenarios.

2 Related Work

The existing representative research on learning path recommendation can be divided into three categories, based on learner characteristics, semantics and cognitive relations between knowledge units.

2.1 Learning path recommendation based on learner characteristics

Learning path recommendation based on learner characteristics generates learning path by analyzing characteristics of the learning behavior observed during learning process. Fan [1] proposed a learning path model that allows learning activities and the assessment criteria of

their learning outcomes to be explicitly formulated by the Bloom's Taxonomy. Chun-Hsiung Lee and Gwo-Guang Lee [14] adopted scaffolding theory [15] and proposed scaffolding learning path algorithm, which digs the learning path of outstanding students to establish a learning navigation path. Xiao Huimin [16] proposed a recommendation method based on particle swarm optimization algorithm. In the proposed method, each learner is considered as a particle to make a personalized learning path optimization. Yan Cheng [17] proposed a learning path recommendation method based on swarm intelligence. The learning path is selected based on the pheromone contribution from the adjacent user. Berg [18] argued that if a learning path is used frequently, then there is a high probability that learner will use this path to learn again. Chun [19] presented a fusion of personalized learning and game based learning. They developed a personalized innovative learning system based on decision tree to provide personalized learning path for learners. Basu [20] presented a user model based system which takes into account a member of parameters such as learner's preference, previous performance, requirement of credit points, and availability of time to recommend a learning path. Bendahmane [21] presented a competence based approach (CBA) based on learning data, learner's characteristics and their expectations. In this approach, learners were clustered and traced, and finally proper learning paths were presented. Salehi [22] introduced learner preference tree (LPT) in which the multidimensional attributes of material, rating of learners, ordered and sequential patterns of the learner's accessed material were put together into consideration. The model uses mixed, weighted, and cascade hybrid methods to form the final recommended learning paths.

The above learner-based methods mainly recommend learning paths from the learner's point of view. They are based on the parameters such as a learner's preference, his/her previous performance, learning targets, and learning abilities etc. There is a tendency that e-learners often use time fragments to learn and it led to their fragmented and inconsecutive learning behaviour. However this increases the difficulty of mastering the knowledge. So a consecutive learning path is preferred if the continuity of the knowledge elements is considered. The learner-based modeling methods have partly been taken into account the order of the knowledge elements in the learning process, yet it is impossible to ensure that the knowledge units are provided continuously systematically.

2.2 Learning path recommendation based on semantic relations

Learning path recommendation is essentially different from the general merchandise recommendation and the movie recommendation. The key factors that affect the results of commercial product recommendation mainly depend on the user's evaluation, rating, browsing,

or collection which has limited relationship with the user's understanding and cognition. On the other hand, the massive educational resources have some inherent semantic structure characteristics, which determines that the knowledge elements have logical sequences [12]. According to the theory of connectivism, learning is a process of constantly connecting knowledge nodes or resources. The internal relationship between knowledge elements has an important role in learning process [23].

Kuo-Kuan Chu [24] et al proposed a learning path generation algorithm based on ontology. In this method, the ontology base is established based on the relationships between knowledge elements, and then a learning path is recommended according to the relationship between them in the ontology. Massimo [25] proposed a method based on Bayesian network to generate learning path. They used the concept of the domain ontology to map the learning path recommendation to a sort of constraint satisfaction problem. Chih-Ming Chen [26] recommended learning paths focusing on difficulty of learning materials and learning ability. And their proposed approach is based on item response theory. Yang [27] adopted a self-organized rule to organize learning contents as a multi-faceted semantic link network, and then they used the inference engine to generate the recommendation for the learning path which matches the user's learning style.

The key underpinning concepts of the above methods are to perform the semantic analysis to form a concept map among knowledge units. The advantage of the recommendation methods based on concept map is that they consider the inherent relation between knowledge units, but most of these methods failed to consider the learning sequence between connected knowledge units. Therefore, these approaches ignore the precision required for the effective recommendations.

2.3 Learning path recommendation based on cognitive relations

Most of the above learning path generation algorithms seldom consider the target knowledge units, and they ignore the impact of the cognitive relationship between knowledge units. Recent years has seen increased from industry and academia to construct resources' semantic description [28] for effectively extracting structured contents from massive web data. The massive and heterogeneous learning resources lead to knowledge disorientation and cognitive overload for the learners. To overcome this difficulty, knowledge map [12] is put forward as a novel learning resource organization mode. In a knowledge map, knowledge units in a course or a subject are organized as a big graph.

The knowledge map has been applied in knowledge management, storage, learning

navigation, and so on [29][30]. Ferran [31] used knowledge map to manage the learning contents. Ria [32] showed how to develop lifelong learning skills by creating a digital cooperation knowledge map. Liu [12] found the relationship between the knowledge elements from the text-based learning resources. A knowledge map is used to analyze the topological relationship. Lin JC [33] obtained the successful case of groups to make a path recommendation to the other learners through mining logs based on Knowledge Map. Guillaume Durand [34] proposed a learning path recommendation system based on graph theory, and a greedy algorithm is used to find local optimal solution for the shortest path. A large number of empirical studies in the field of cognitive science and network learning have also proved that the learning environment based on knowledge map is better than the text and the traditional learning environment[35][36]. Wan [37] paid attention to learners' emotions and proposed a learner oriented recommendation approach based on mixed concept mapping and immune algorithm (IA). He modeled the learner oriented recommendation as a constraint satisfaction problem (CSP) which aims to minimize the penalty function of unsatisfied indexes. Tam [38] considered that the course instructor's views on the relations of the involved concepts/modules can be imprecise or even contradictory and then presented a new e-learning system which could perform an explicit semantic analysis on the course materials to extract the individual concepts. The results combined with the views of experts were used to form final learning paths.

These learning path recommendation methods mentioned above are mainly based on knowledge map according to cognitive order. The sequences of the knowledge units have been well considered when the knowledge maps are formed. Thus recommended learning paths will meet the general needs of the learners when they are learning knowledge. Their work requires further improvement when considering the diversity of learning scenarios. Learners will put forward different requirements under different learning scenarios. An ideal recommendation algorithm should consider learners' learning process and learning abilities etc., and then provides diversified learning paths to different learning scenarios.

However, after conducting experiment, we have found that learners have a great randomness when they were learning through knowledge map. In this experiment, GSP algorithm [39] is used to find the frequent sequential pattern of knowledge elements based on single learning duration. The raw data consists of 2150 log items from 156 learners and 57 knowledge units. The mining results are $\{Seq_i\} = \{<\#1, \#2>, <\#1, \#8>, <\#2, \#3>, <\#1, \#3>, <\#1, \#6>, <\#2, \#4>, <\#2, \#7>, <\#6, \#8>, <\#8, \#5>, <\#3, \#6, \#8, \#9>, <\#3, \#4, \#6, \#9>, <\#5, \#6, \#8, \#9>, <\#1, \#4, \#6, \#10>, \dots\}$ when setting the value of support [40] as 0.1. Obviously, we can find that learning dependency from a previous knowledge unit to its latter knowledge unit

in each sequence, Seq_i , is mostly missed in the learning sequence, when comparing Seq_i with its corresponding partial paths in the context of the knowledge map as shown in Figure 1. This can cause a great difficulty for learners to master the knowledge, especially when they cognitively make newly-learned knowledge units connected to the knowledge units they already mastered. To recommend appropriate learning path for learners to solve the problems mentioned above is one of the core research in this paper.

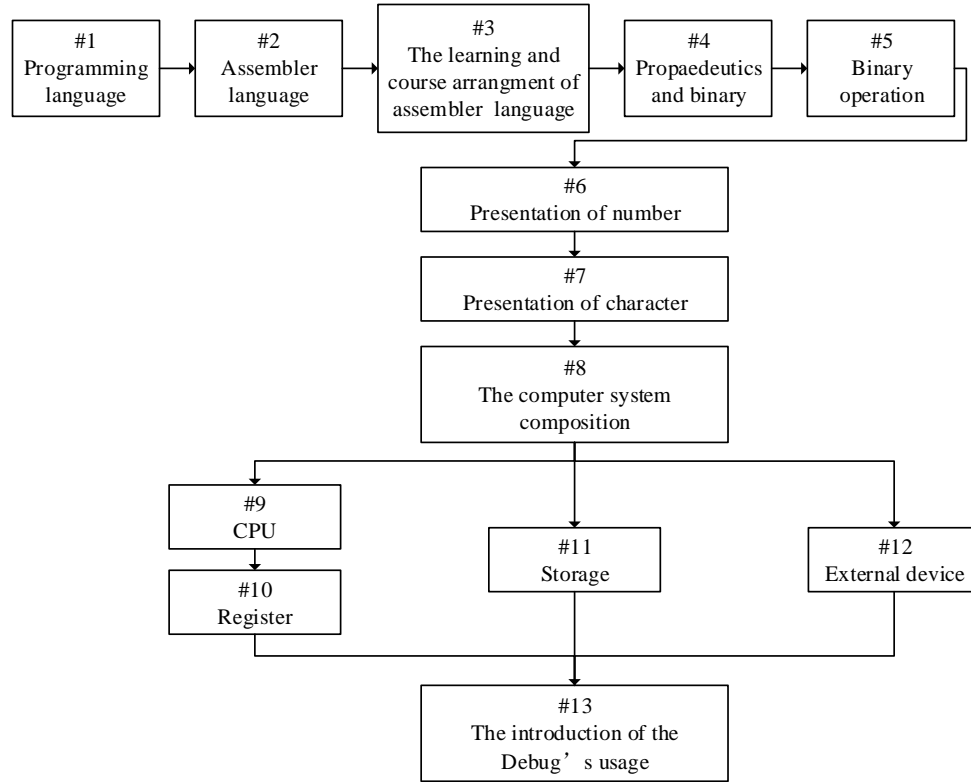


Figure 1 Part KM of the course Assembly Language

Based on learner's prior knowledge and learning goals, using knowledge map as unit of knowledge structure description tool, this paper proposes a multi-constraint learning path recommendation algorithm, which offers learning path recommendation according to the current learning scenarios and learning goals.

3 Multi-constraint Learning Path Recommendation Algorithm

In this section, firstly, we give five definitions we will use in the following sections. Secondly, we explain two indexes of knowledge unit which represent attention degree and the learning centrality of KU. Thirdly, the multi-constraint model of learning path is proposed. Finally, according to the model, we design and realize the multi-constraint learning path recommendation algorithm.

3.1 Related definitions

Definition 1: Knowledge Unit (KU): KU is a complete knowledge expression that cannot be further divided [41]. For example, “Java constant” is a knowledge unit of the course on Java Programming.

Definition 2: Target Knowledge Unit: A target Knowledge Unit is a knowledge unit that a learner wants to learn in a learning process. There may be several paths from the start knowledge unit to the target knowledge unit.

Definition 3: Learning Dependency: It refers to a kind of necessary dependency relation between knowledge units during a learning process [42]. Figure 2 shows an example of learning dependency between three knowledge units.

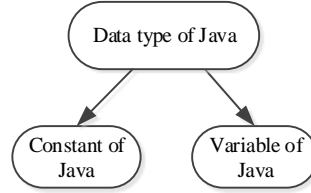


Figure 2: example of learning dependency between knowledge units

Definition 4: Knowledge Map (KM): Knowledge map is a directed graph that regards knowledge units as points/nodes and the relationships between knowledge units as edges. Knowledge map is denoted by

$$KM = (KU, KE) \quad (1)$$

in which:

KU is a set of all knowledge units in knowledge map, KE is a set of all relations between knowledge units in a knowledge map.

The adjacency matrix of the knowledge map KM is referred as $C = (c_{ij})_{n \times n}$, $0 < i \leq n$, $0 \leq j < n$. $i \neq j$ C satisfies formula 2.

$$\begin{cases} c_{ij} = 1 & (ku_i, ku_j) \in KE \\ c_{ij} = 0 & (ku_i, ku_j) \notin KE \end{cases} \quad (2)$$

ku_i is a specific knowledge unit of knowledge map, which is the basic knowledge unit with complete expression ability. $ku_i \in KU$

ke_i is a learning dependency relation in the knowledge map, which refers to a relationship between pre order relation, causality and case relation, and so on. $ke_i \in KE$

Definition 5 Learning path: learning path is a sequence of multiple knowledge units which is determined by the target knowledge unit, and denoted by $p = \{ku_i, ku_j, \dots, ku_m\}$, in which $ku_i, ku_j, \dots, ku_m \in KU$.

In knowledge map, there would be multiple learning paths from the initial knowledge unit to the target knowledge unit. The goal of this paper is to recommend learning path satisfying multiple constraints based on the learner's learning log.

3.2 Attention degree of KU

Taking Assembly Language course as an example, we have selected 156 learners and have analyzed the total learning frequency and total learning duration for 57KUs. The result is shown in Figure3.

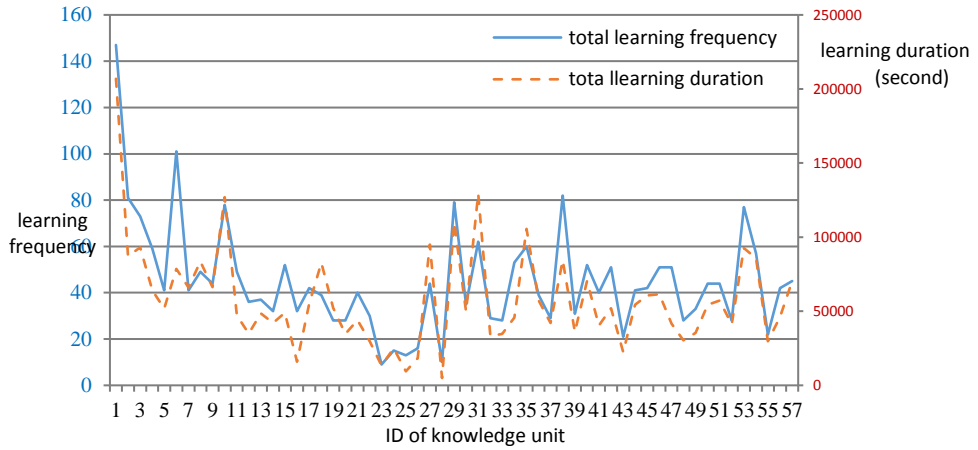


Figure 3 Total learning frequency and total learning duration for each KU of Assembly Language

As presented in Figure 3, some KUs' total learning frequency and total learning duration is significantly higher than other KUs. Therefore, we define attention degree as a measure of the learner's interest in a KU. For a KU, its attention degree is actually the ratio between all learners' average learning duration and the original duration of the KU itself.

$$h_i = \frac{T_{isum}}{F_{isum} * t_{i0}} \quad (3)$$

T_{isum} is the total learning duration of ku_i , F_{isum} is the total learning frequency of ku_i and

is the original duration of ku_i .

3.3 Learning centrality of KU

From the view of KUs' semantic inheritance, some KUs play a core role in the learning process, and they have high semantic contribution ability to the other KUs' in a learning process. So when a critical learning path recommendation is proposed, the key KUs should be taken into consideration. Based on our team's previous research [13], an absorbing state Markov model for the semantic migration of KM is constructed, and then the KU's learning centrality d is used to describe the statistical feature of the degree of the importance of KU's.

To calculate KU's learning centrality, we set the current KU as ku_i . q_{ij} is define as the probability of the semantic information migrated to its cognitive consequent ku_j . The KU's development potential shows its degree of being semantically inherited, which is defined as the KU's out-degree e_i^{out} . So for ku_i , the larger out-degree means the more cognitive consequents, meanwhile, probability of each candidate cognitive consequent to be chosen is smaller. In the KM, e_i^{in} is defined as the KU's in-degree. The larger in-degree means the more cognitive antecedents, which leads to hard to learn that KU. After the calculating probability of the semantic migration, q_{ij} , we obtain the absorbing state Markov model for the semantic migration of KM. Thereafter the KU's learning centrality is calculated.

3.4 Multi-constraint model of learning path

It is imperative that not only the learners' targets but also the learning scenarios should be considered during a learning process [43]. In this paper, we proposed four basic learning scenarios based on the time when the learner starts learning and the learning status of the target knowledge unit.

- a. Initial Learning: First time study of target KU in ordinary study means that the learners study the target KU first time, and there is an adequate time for study.
- b. Ordinary Review: Reviewing the target KU in ordinary study means that the learners' ordinary review of the knowledge that they have already studied, and there is adequate time for reviewing.
- c. Pre-exam Learning: First time for studying the target KU just before the exam means

the learners study the target KU first time, and there is not much time for study.

- d. Pre-exam Review: Reviewing the target KU just before the exam means that the learners review those knowledge units they had already studied, and there is not much time for study.

3.4.1 Requirement analysis of multi-constraint learning path

Based on the learners' log and the four kinds of learning scenarios presented above, we proposed seven kinds of scenario-oriented requirements for the learning path according to characteristics of learners and of knowledge map.

1) Complemented learning path

It is a learning path in which the learner's goal graph contains most KUs which have not been effectively learned. Here the user's goal graph means knowledge sub-graph which is generated by the paths from the starting KU to the target KU. Note that, a KU is considered to be learned effectively, if and only if its total learning duration is more than 80% of its original duration.

2) Shortest learning path

The learning path in which the learner's goal graph contains the least number of KUs.

3) The shortest duration learning path

The learning path in which the learner's goal graph contains the shortest total duration of KUs.

4) Critical learning path

The learning path in which learner's goal graph contains maximum learning centrality of KUs. The learning centrality is one attributive character of the knowledge unit, and it is used to measure the importance of a KU in the whole knowledge map of a course.

5) Easy learning path

The learning path in which the learner's goal graph contains the KUs of highest learning frequency. The more learning frequency of a KU, the more learner is familiar with the KU. Total frequency of the KUs is used as the measurement to determine whether the learning path is easy to learn or not.

6) Complete learning path

The learning path in which the learner goal graph has more KUs that have not been learned.

7) More-hotspot learning path

The learning path in which the learner goal graph contains the KUs which have the highest attention degree. The attention degree of KU is another learner's related features, which is used to measure the learner's interest degree of the course's KUs.

In these seven paths, the complemented learning path, easy learning path, complete learning path and more-hotspot learning path, are proposed based on the learner's characteristics in learning log. The learning path with shortest duration, shortest learning path and critical learning path are based on the inherent attributes of the course's knowledge map.

3.4.2 Construction of a multi-constraint learning path recommendation model based on knowledge map

According to the seven kinds of learning path requirements, formal representations of these learning paths' constraints factors are shown in Table 1.

Table 1 learning paths' constraints factors

learning path	Constraint factor	Parameters
repair learning path	$f_1 = \frac{n_r}{m_r + 1}$	n_r is the number of KUs which are not effectively learned in the learner goal graph. m_r is the number of KUs which are not effectively learned in a typical learning path.
shortest learning path	$f_2 = \frac{l}{l_M}$	l represents number of KU of a learning path, which defines path length. l_M is the longest path length of the learner's goal graph.
shortest learning duration path	$f_3 = \frac{l_t}{l_{t_M}}$	l_t is the duration of a path, namely, the total video duration of KUs in a path. l_{t_M} is the longest duration of a path in a learner's goal graph.
critical learning path	$f_4 = \frac{l_{d_M}}{l_d}$	l_d is the sum of KUs' learning centrality of a path. l_{d_M} is the biggest/longest l_d of the path in a learner's goal graph.
easy learning path	$f_5 = \frac{l_{w_M}}{l_w}$	l_w is the total learning frequency of KUs in the path. l_{w_M} is the biggest l_w of the path in a learner goal graph.

complete learning path	$f_6 = \frac{n_u}{m_u + 1}$	n_u is the number of KUs which are not learned in the learner goal graph. m_u is the number of KUs which are not learned in the learning path.
more-hotspot learning path	$f_7 = \frac{l_{h_M}}{l_h}$	l_h is the sum of the KUs with more attention degree. l_{h_M} is the biggest l_h of the path in the learner goal graph.

To satisfy the learner's diversity of learning scenario, the multi-constraint model of learning path recommendation is constructed as shown in Formula (11):

$$f = \alpha * g(f_1) + \beta * g(f_2) + \chi * g(f_3) + \delta * g(f_4) + \gamma * g(f_5) + \lambda * g(f_6) + \zeta * g(f_7) \quad (4)$$

In Formula (11), g is a function for calculating Min-Max standardization of the seven learning paths' constraint factors. α , β , χ , δ , γ , λ , ζ are the constraints for recommending learning paths for learners, which are the weighted coefficients of the constraint factors. By adjusting the weighted coefficients, we can gain different learning path in different constraint conditions. For example, when $\alpha = 1, \beta = 0, \chi = 0, \delta = 0, \gamma = 0, \lambda = 0, \zeta = 0$, the number of KUs which are not effectively learned in the learning path are less, and the value of f is smaller, and this situation is suitable for the repair learning path. Similarly remaining six paths can be calculated. When $\alpha = 0, \beta = 0.5, \chi = 0, \delta = 0, \gamma = 0.5, \lambda = 0, \zeta = 0$, the learning path l is shorter, the total learning frequency of the KUs in the path is bigger, and the value of f is smaller, and this situation is suitable for the shorter length and easier learning path, which is a quick learning path that satisfies two constraint factors and it is suitable for the Distance Learning College learners(DLC-learners). We use Φ to denote the constraint conditions, $\Phi = \{\alpha, \beta, \chi, \delta, \gamma, \lambda, \zeta\}$.

3.5 Design and realization of the multi-constraint learning path recommendation algorithm

In this section, we present design and realization of the multi-constraint learning path recommendation algorithm.

3.5.1 Design of multi-constraint learning path recommendation algorithm based on KM

The architecture of the multi-constraint learning path recommendation algorithm is

```

graph TD
    KM[Knowledge map] --> LEP[Log extraction and preprocessing]
    LL[(Learning log)] --> LEP
    LEP --> G[Generate the user learned G]
    G --> AM[Store as the adjacent matrix]
    AM --> F1[Find the paths between (S, E) in G]
    F1 --> F2[Construct the learner goal graph G']
    F2 --> F3[Calculate the constrained factor f]
    F3 --> RLP[Generate the recommended learning path]
    RLP --> RLP_Sequence((ku1 -> ku3 -> ku5 -> ku6))
    
```

The description of the algorithm is as follows:

- The KM's adjacent matrix shows the learning dependency between the KUs. For a directed KM which includes n KUs, a $n \times n$ matrix $C = (c_{ij})_{n \times n}$ is defined. If C satisfies the formula (2), we call it as the adjacent matrix of the course's KM.

- According to the users' learning log, we mark the studied knowledge units in knowledge map and get the user learned graph, denoted as $G(id, t)$. A weight adjacent matrix is used to represent $G(id, t)$. The weights include the learned label, the original duration of the KU video, the learning centrality of KU, the learning frequency of KU and the attention degree of KU.

$$G(id, t) = \{ku, ke, (r, t_{i_0}, d, w, u, h)\} \quad (5)$$

14

learner's learning frequency. u is the learner's learning label of ku , $u = 1$ means ku has already been learned, while $u = 0$ means ku has not been learned. h is the attention degree of ku . The generation algorithm of the user's learned graph is shown in algorithm1.

Algorithm 1 Generation algorithm of the user's learned graph

Input: the adjacent matrix of the KM $C = (c_{ij})_{n \times n}$ and the learner's learning log $\{UserLog_{id}\}$	
Output: user's learned graph $G(id, t)$	
<hr/>	
1: for all $ku_i \in KU$ do	
2: for all $\{UserLog_{id}\}$ do	
3: $r = getR();$	{label if ku_i have already been effectively finished }
4: $t_{i0} = getT();$	{get the original video duration of ku_i }
5: $d = calculateD();$	{calculate the learning centrality of ku_i }
6: $w = calculateW();$	{calculate the learning frequency of ku_i }
7: $u = getU();$	{ label if ku_i has already been learned }
8: $h = calculateH();$	{calculate the KUs' attention degree}
9: end for	
10: end for	
11: return $G(id, t)$;	{return the user learned graph}

3) Constructing user's goal graph

All paths from the start KU S to the end KU E in $G(id, t)$ are produced based on the depth-first traversal algorithm. Then we can construct the user's goal graph $G'(id, t, S, E)$.

Algorithm 2 Generation algorithm of the user's goal graph

Input: the adjacent matrix of the KM $C = (c_{ij})_{n \times n}$, the user's learned graph $G(id, t)$, the start KU S , the target KU E	
Output: all paths between (S, E)	
<hr/>	
1: $pathf(s, e)$	{define function}

```

2: if  $s == e$  then
3:   return  $path$ ;           {if start node and end node are the same, then return the path}
4: else if  $s$  connect to  $e$            {if there is directly connected edge between the two nodes}
5:   if  $andmark == 1$  then {if the relationship between two nodes is AND relationship}
6:     return;           {Back to the other side of the AND relationship.}
7:   else
8:      $pathf(v, v)$ ;           {recursively call the function  $pathf$ }
9:   end if
10: else           {if there is no connection between the two nodes}
11:   for all  $w$  connect to  $s$  do
12:      $pathf(w, e)$ ; { if there is no direct connected edge between the two nodes, then find the paths of
                        nodes  $w$  and  $e$ , which are adjacent to nodes}
13:   end for
14: end if
15: }
16: return  $all\ path$            {return all paths between  $(s, e)$ }

```

4) Generating constraint learning path

According to the constrained condition $\Phi = \{\alpha, \beta, \chi, \delta, \gamma, \lambda, \zeta\}$ and the multi-constraint model in formula (10), all values f of all the learning paths in the user's goal graph $G'(id, t, S, E)$ are calculated. Then we get the learning path recommendation $P(S, E, G', \Phi)$, which satisfies all constrained conditions. The generation algorithm of the constrained learning path is shown in algorithm 3.

Algorithm 3 Generation algorithm of the constrained learning path

Input: the user's goal graph $G'(id, t, S, E)$, constrained condition $\Phi = \{\alpha, \beta, \chi, \delta, \gamma, \lambda, \zeta\}$

Output: the constrained learning path recommendation $P(S, E, G', \Phi)$

```

1: for all  $path$  do

```

```

2:  $f_1 = n_r / (m_r + 1)$ ;      { calculate the constraint factor of the complemented learning path }
3:  $f_2 = l / l_M$ ;              { calculate the constraint factor of the shortest learning path }
4:  $f_3 = l_t / l_{t_M}$ ;          { calculate the constraint factor of the shortest duration learning path }
5:  $f_4 = l_{d_M} / l_d$ ;          { calculate the constraint factor of the critical learning path }
6:  $f_5 = l_{w_M} / l_w$ ;          { calculate the constraint factor of the easy learning path }
7:  $f_6 = n_u / (m_u + 1)$ ;          { calculate the constraint factor of the complete learning path }
8:  $f_7 = l_{h_M} / (l_h + 1)$ ;      { calculate the constraint factor of the more-hotspot learning path }
9:  $f = \alpha * g(f_1) + \beta * g(f_2) + \chi * g(f_3) + \delta * g(f_4) + \gamma * g(f_5) + \lambda * g(f_6) + \zeta * g(f_7)$ ;
10: end for
11: for all path do
12: if  $f = \min(f)$  then          { path which has the smallest  $f$  is the constrained learning path }
13: return  $P(S, E, G', \Phi)$ 
14: end if
15: end for

```

3.5.2 Realization of the multi-constraint learning path recommendation algorithm

This paper is uses the KM of Assembly language course from the DLC of our school. Appendix A shows the original video duration, the calculation results of the attention degree and the learning centrality of each knowledge unit in Assembly language course. Let “binary operation” be the start KU S , and “pseudo-operation” be the target KU E . The learned KUs’ weight information in (S, E) of the No. 0035 learner is shown in Table 2.

Table 2 characteristic values of the learned KUs in (S, E)

Sequence number of KU i	Effectively finished sign r	original video duration t_{i0} (second)	Learning centrality d	Learning frequency w	Learning label u	attention degree h
#5	1	603	2.2457	6	1	1.0963

#8	1	997	2.9754	11	1	1.7119
#9	1	1298	3.4665	3	1	1.1591
#10	0	1674	4.6880	2	1	0.9719
#11	0	1317	1.4234	2	1	0.7323
#14	1	619	1.5728	11	1	2.0063
#15	1	719	1.6554	3	1	1.3100
#19	1	1040	1.9008	4	1	1.7901
#22	0	1251	2.4341	5	1	0.7913
#23	1	1382	2.6987	7	1	1.0757

According to the KM of Assembly Language and the user's learned graph, the user's goal graph can be constructed. And finally, we generated a multi-constraint learning path, which is shown in Table 3.

Table 3 Results of multi-constraint learning paths

Constrained conditions Φ	Path name		$P(\#5, \#30, G', \phi)$
$\{1, 0, 0, 0, 0, 0, 0\}$	Complemented	learning path	{binary operation, the presentation of number, the presentation of character, the computer system composition, external device, the introduction of the Debug's usage, instruction system and addressing mode, data transfer instruction, arithmetical instruction, logic instruction, String processing instruction, control transfer instruction, processor control instruction, pseudo-operation1, pseudo-operation2}
$\{0, 1, 0, 0, 0, 0, 0\}$	Shortest	learning path	{binary operation, the presentation of number, the presentation of character, the computer system composition, instruction system and addressing mode, directly addressing mode, pseudo-operation 1, pseudo-operation 2}
$\{0, 0, 1, 0, 0, 0, 0\}$	Shortest	learning duration path	{binary operation, the presentation of number, the presentation of character, the computer system composition, instruction system and addressing mode, the addressing mode 1 of the operand in the RAM, the addressing mode 2 of the operand in the RAM, the addressing mode of the instruction jump, pseudo-operation1,

		pseudo-operation 2}
{0,0,0,1,0,0,0}	Critical learning path	{binary operation, the presentation of number, the presentation of character, the computer system composition, CPU, register, valid address, data transfer instruction, stack and its instruction, input and output instruction, pseudo-operation1, pseudo-operation 2}
{0,0,0,0,1,0,0}	Easy learning path	{binary operation, the presentation of number, the presentation of character, the computer system composition, CPU, register, instruction system and addressing mode, data transfer instruction, stack and its instruction, input and output instruction, pseudo-operation1, pseudo-operation 2}
{0,0,0,0,0,1,0}	Complete learning path	{binary operation the presentation of number, the presentation of character, the computer system composition, external device, the introduction of the Debug's usage, instruction system and addressing mode, data transfer instruction, stack and its instruction, input and output instruction, arithmetical instruction, logic instruction, String processing instruction, control transfer instruction, processor control instruction, pseudo-operation1, pseudo-operation 2}
{0,0,0,0,0,0,1}	hot pot learning path	{binary operation, the presentation of number, the presentation of character, the computer system composition, CPU, register, instruction system and addressing mode, direct addressing mode, register addressing mode, pseudo-operation1, pseudo-operation 2}
{0,0.5,0,0,0.5,0,0}	quick learning path	{binary operation, the presentation of number, the presentation of character, the computer system composition, instruction system and addressing mode, data transfer instruction, stack and its instruction, input and output instruction, pseudo-operation 1, pseudo-operation 2}

4 Experiment result and analysis

This section describes our experiment, its design and results to verify the hypotheses we proposed.

4.1 Design of the experiment process and questionnaire

This section describes the experiment process and the questionnaires results that we used to verify the hypotheses.

4.1.1 Design of the experiment process

An experiment process for verifying the hypotheses and the recommendation algorithm of learning paths is shown as Figure 5.

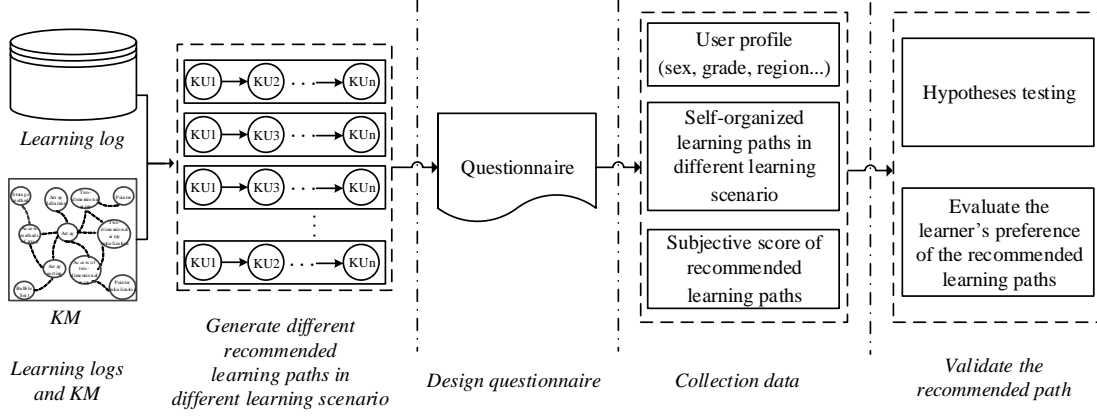


Figure 5 Experiment process for verifying the hypotheses and the recommendation algorithm of learning paths

As shown in Figure 5, firstly, using the KM and learning logs, we generate different recommended learning paths in various learning scenarios, which will be used in the questionnaire. Secondly, we design a questionnaire, in which learners' basic information, self-organized learning paths in different learning scenarios and subjective score of the eight recommended learning paths in every learning scenarios can be collected. Thirdly, we send out questionnaires and then collect the data. Finally, we verify the hypotheses we proposed and evaluate the learners' preference of the recommended learning paths.

4.1.2 Design of the questionnaire

The questionnaire includes three parts. The first part is used to collect some basic information about the learners. The second part enables learners to give their self-organized learning paths from the start KU S, #5, to the target KU E, #30, in different learning scenarios, and the content of the second part is shown in Table 4. The third part enables the learners to select their subjective score of the eight recommended learning paths in every learning scenarios. We use five-point scale. Five points represent very satisfied, four points means satisfied, three points means general, two points means unsatisfied, as well as one point means very unsatisfied. Table 5 presents an example of learners' subjective scoring to the complemented learning path.

The graph in Table 5 is a partial knowledge map of a course 'Assembly Language', whose

detailed information is as follows: set the start KU S as #5(the operation of binary) and target KU E as #30(pseudo-operation 2). A learning record from S to E is represented as {#5(2), #8(11), #9(3), #10(2), #11(2), #13(14), #14(11), #19(4), #22(5), #23(7)}, in which #5(2) means the KU ID is 5, and the learning frequency is 2.

Table 4 Learners' self-organized learning paths in four different scenarios

Learning scenario	autonomous learning path
1first learn(E has not been learned)	
2 usual review(E has been learned)	
3pre-exam learn(E has not been learned)	
4pre-exam review(E has been learned)	

Table 5 Learners' subjective scoring of the recommended learning paths in four learning scenarios

1 the complemented learning path (which includes most unlearned KUs)	first learn	usual review	pre-exam learn	pre-exam review
#5 #6 #7 #8 #12 #13 #14 #21# 24 #25 #26 #27 #28 #29 #30	score	score	score	score
	(1-5)	(1-5)	(1-5)	(1-5)

4.2 Experiment

In this part, firstly, we analyze some basic information about 110 learners. Secondly, two hypotheses we proposed are verified. Then, we evaluate the learners' preference of the recommended learning paths. Based on the evaluation results, we conclude the strategy of the recommended learning path.

4.2.1 Experiment description and object basic information analysis

The experiment carried out has 110 questionnaires filled in and of which 105 questionnaires were valid. Figure 6, 7 and 8 show some of detailed information about 110 DLC-learners. The number of Batch in Figure 7 means DLC-learner's admission time.

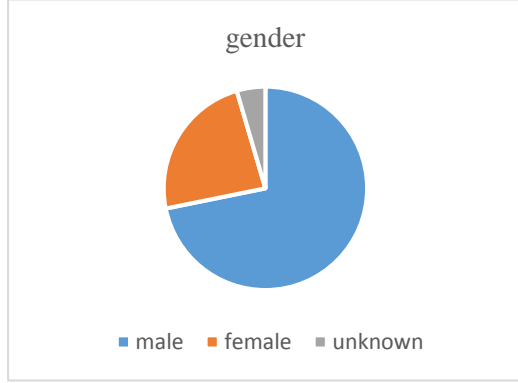


Figure 6 Learners' gender information

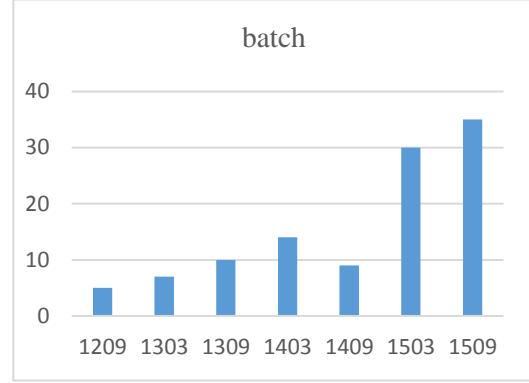


Figure 7 Learners' batch information

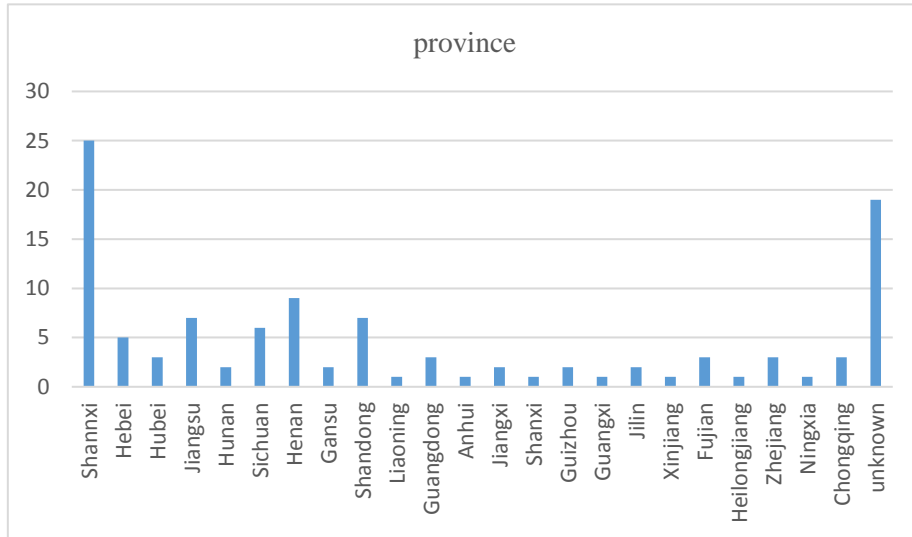


Figure 8 Learners' region information

4.2.2 Hypothesis testing

In this section, we verify the two hypothesis we proposed in the introduction section, the first one is that a learner has different learning path requirements in different learning scenarios and the second is that different learners have convergent learning path requirements in the same learning scenario.

We use Edit Distance algorithm [43] (see Appendix B) to calculate the similarity between the learners' self-organized learning paths and the recommended eight learning paths. When the value of EditDistance is zero, it means that the two paths are identical, otherwise the

bigger the value of the EditDistance of the two paths is, the larger difference between the two paths is.

For each learner, we calculate all the EditDistance between his/her four self-organized learning paths and the eight recommended learning paths. Table 6 shows the calculation result of a learner.

Table 6 All the EditDistance between of a learner's four self-organized learning paths and the eight recommended learning paths

	slef-organized learning path in first learn	slef-organized learning path in usual review	slef-organized learning path in pre-exam learn	slef-organized learning path in pre-exam review
complemented learning path	0	5	5	5
shortest learning path	5	4	0	2
shortest duration learning path	5	4	2	2
critical learning path	5	4	0	2
easy learning path	5	0	4	4
complete learning path	0	5	5	5
more-hotspot learning path	5	4	2	2
quick learning path	5	4	2	0

● Verification of Hypothesis I

According to the result shown above, for each learner, we take one of the eight recommended learning paths, which has the smallest EditDistance in all of the eight paths, as his/her required learning path in one specific learning scenario. Here are some examples of learners' requirements in different learning scenarios. In Table 7, P1 to P8 refer to the mentioned eight kinds of constraint learning paths

Table 7 Example of learners' path requirements in different learning scenarios

	first learn	usual review	pre-exam learn	pre-exam review
learner A	P6	P5	P4	P4
learner B	P4	P6	P2	P8
learner C	P6	P4	P4	P7

We proposed an indicator, diversity of paths (DoP) to represent the diversity of a learner's choices of learning paths in different learning scenarios. If four paths are completely different, the value of DoP is 4. If and only if two in the four paths are identical, the value of DoP is 3. If there exist only two different paths in the four paths, the value of DoP is 2. And if all the four paths are identical, the value of DoP is 1. For example, in Table 6, DoP of learner A and learner C is 3, DoP of learner B is 4. In Formula 13, function *UNIQUE()* is used to find the distinct choices (learning paths) of one learner's choices, and function *LENGTH()* is used to calculate the number of the distinct choices.

$$\text{DoP} = \text{LENGTH}(\text{UNIQUE}([\text{learner's choices in different learning scenarios}]))$$

(13)

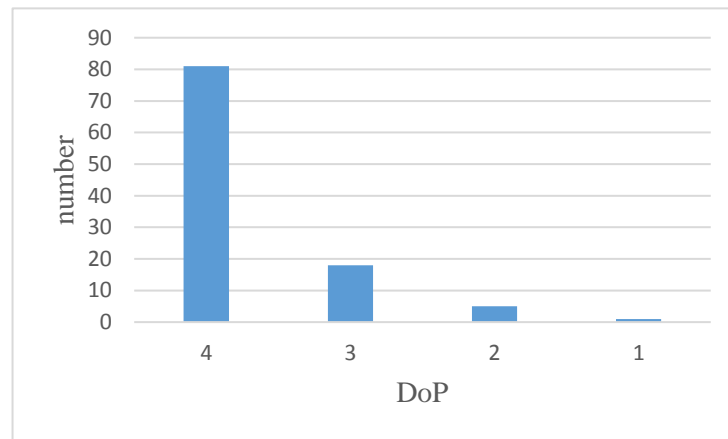


Figure 9 Number of learners of different DoP We find that 77% learners' DoP is 4, and 17% learners' DoP is 3. Therefore, we conclude that most of the learners have different learning path requirements in different learning scenarios.

● Verification of Hypothesis II

For each kind of recommended learning paths, we count the number of learners who choose it in the four learning scenarios. The result is shown in Figure 10.

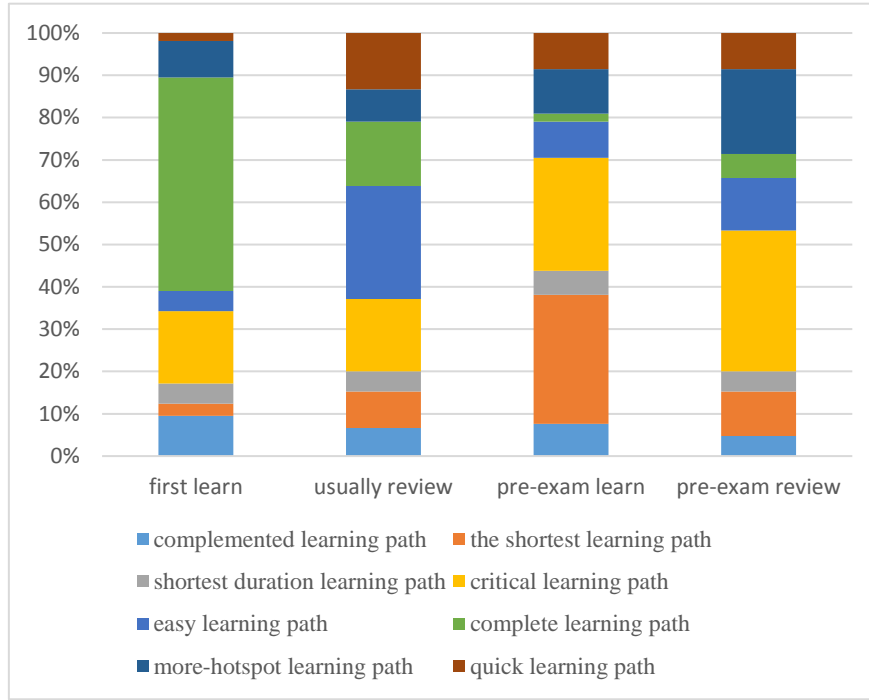


Figure 10 Learners' requirements of learning paths in the same learning scenario

As can be seen from Figure 10, in the first learning scenario, most learners chose complete learning path, which means that they wanted to learn more KUs that they had not learned. In the usual review scenario, the learners preferred an easy learning path, but the difference among the eight learning paths is not obvious. In pre-exam learning scenario, the learners liked the critical and the shortest learning paths, namely, they wanted to learn the important KUs or just chose the shortest path for learning KUs. In the pre-exam review scenario, the learners preferred the critical and more-hotspot learning paths, which is probably because they wanted to learn the important or hot KUs. Therefore, it can be concluded that different learners have convergent learning path requirements in the same learning scenario.

Moreover, we perform correlation analysis between learners' profiles, focusing on gender, age and province, and their own learning path preference through the chi-square test. For example, in usual review, we perform correlation analysis between gender, path preference, and the result of the chi-square test is $\chi^2 = 4.737$. For $df = 7$ and $\alpha = 0.05$, $\chi^2_{0.05} = 14.067$, so obviously learners' gender and path preference is independent in usual review. Meanwhile, all results show that the learners' profiles are independent of the learner's path preference. So the learners' profiles have not been considered in the learning path recommendation.

4.2.3 Evaluation of learners' preference to the recommended paths

By calculating the similarity between the 105 learners' self-organized learning paths and the recommended eight learning paths in four learning scenarios, we found that 87.6 percent of

105 learners had given the highest score or the second highest score to the recommended learning paths, also the edit distance between the self-organized learning path and one of the recommended learning paths is the smallest. So according to the statistical similarity of the learners' self-organized learning path and the recommended learning path, we proposed the learner's preference index Ω of the learning paths.

$$\Omega = Score * \frac{Length\ of\ Autonomously\ Path}{EditDistance} \quad (6)$$

In formula 6, for one learning scenario, the parameter *score* is a learner's subjective score to one of the recommended learning paths, *Length of Autonomously Path* is the length of the learner's self-organized learning path and *EditDistance* is the EditDistance between the self-organized learning path and the recommended learning path.

We calculated each learner's preference Ω to the eight recommended learning paths in each learning scenarios, and then for a learner in one specific learning scenario, we chose the path that had the largest Ω as his/her prefer learning path. Finally, we got the 105 learners' preference learning path as showed in Table 8. For example, in the first row of Table 8, 53 learners chose the complete learning path as their preferred e learning paths.

Table 8 Learners' preference of the learning path

	Complem -ented learning path	the shortest learning path	shortest duration learning path	critical learning path	easy learning path	complete learning path	more- hotspot learning path	quick learning path
first learn	11	2	5	18	5	53	10	1
usual review	7	8	6	15	30	14	9	16
pre-exam learn	10	32	4	29	8	3	15	4
pre-exam review	5	12	10	34	14	5	18	7

Similar to the conclusion drawn from Figure 10, we can also see from Table 8 that learners have certain preference for all eight recommended learning paths in each learning scenarios. Therefore, for a learner in one specific learning scenario, we prefer to recommend one or two

appropriate learning paths according to the result of Table 8.

5 Conclusion

Facing with different learning scenarios and limited time, e-learners need various learning paths to follow, it is necessary to recommend an appropriate learning path to meet their needs and improve the learning efficiency of e-learners. This depends on perceiving the dynamic changes of the learning scenarios in time and making an accurate analysis of the user's fragmentation learning behavior.

The main contribution of the paper is that we proposed a novel approach to overcome the diverse needs of e-learners in different learning scenarios. Based on a knowledge map, the recommended learning path is generated by considering the combination of the domain knowledge structure and cognitive structure of the learners. Firstly, we present four different learning scenarios according to the e-learning process. Secondly, considering e-learner's different path requirements in different learning scenarios, eight kinds of constraint learning paths and their corresponding constraint factors are presented based on the characteristics analysis of learner and resources. Thirdly, a multi-constraint learning path recommendation algorithm based on knowledge map for different learning scenarios is proposed. Finally, the experiments verified the statistical similarity of the learners' self-organized learning path and the recommended learning path in the four learning scenarios. We can draw conclusion that the proposed algorithm is effective for e-learners.

Additionally, there are some limitations that require further improvement. Currently we have only considered four basic learning scenarios according to the learning process. In future work, scenarios could be divided into fine-grained scene according to user's demands. Furthermore, we will optimize the multi-constraint learning path recommendation model on basis of the learner's feedback, and then provide more personalized and more accurate learning paths for e-learners.

References

- [1] Fan Yang, Zhenghong Dong. Learning Path Construction in e-Learning[M]. Springer Singapore, 2017.
- [2] Advanced Distributed Learning. SCORM 2004 4th Edition Version 1.1 Documentation [DB/OL]. <http://www.adlnet.gov/Technologies/scorm/SCORMSDocuments/2004-4th-Edition/Documentation.aspx>.
- [3] Shanshan Wan, Zhendong Niu. A learner oriented learning recommendation approach based on mixed concept mapping and immune algorithm[J]. Knowledge-Based Systems,

- 2016, 103(C): 28-40.
- [4] Chengling Zhao, Zhihui Chen, ZhifangHuang. Recommendation Algorithm and Application of Adaptive Learning Path [J]. China Educational Technology, 2015 (8): 85-91.
 - [5] Gustavo Zurita, Nelson Baloian. Combining learning with patterns and geo-collaboration to support situated learning[C]. IEEE International Conference on Systems, Man, and Cybernetics, 2012: 925–930.
 - [6] Mar Pérez-Sanagustín, Pedro J. Muñoz-Merino, Carlos Alario-Hoyos, Xavier Soldani, Carlos Delgado Kloos. Lessons learned from the design of situated learning environments to support collaborative knowledge construction[J]. Computers & Education, 2015, 87(C): 70-82.
 - [7] Xiaoping Liu. The study of Goal-based scenario(GBS) experiential learning environment[D]. Master degree thesis of Yangzhou University, 2012.
 - [8] Feng Tian, Qinghua Zheng, Zhiyong Gong, Du Jin, Personalized Learning Strategies in e-Learning Environment, Proceedings of the 2007 11th International Conference on Computer Supported Cooperative Work in Design. 2007: 973-978.
 - [9] Kurilovas E, Zilinskiene I, Dagiene V. Recommending suitable learning scenarios according to learners' preferences: An improved swarm based approach[J]. Computers in Human Behavior, 2014, 30:550-557.
 - [10] Hwang G J, Kuo F R, Yin P Y, et al. A Heuristic Algorithm for planning personalized learning paths for context-aware ubiquitous learning[J]. Computers & Education, 2010, 54(2):404-415.
 - [11] Kamsa I, Elouahbi R, Khoukhi F E, et al. Optimizing collaborative learning path by ant's optimization technique in e-learning system[C]. International Conference on Information Technology Based Higher Education and Training. 2016:1-5.
 - [12] Liu J, Jiang L, Wu Z, et al. Mining learning-dependency between knowledge units from text[J]. The VLDB Journal, 2011, 20 (3): 335-345.
 - [13] Xin Wang. A Personalized Recommendation Method of Learning Resource Based on Knowledge Map [D]. Master degree thesis of Xi'an Jiaotong University, 2012.
 - [14] Lee CH, Lee GG, Leu Y. Analysis on the adaptive scaffolding learning path and the learning performance of e-learning[J]. WSEAS Transactions on information science and Applications, 2008, 5 (4): 320-330.
 - [15] Goh J O, Park D C. Neuroplasticity and cognitive aging: the scaffolding theory of aging and cognition[J]. Restorative neurology and neuroscience, 2009, 27(5): 391-403.
 - [16] Huimin Xiao, Caijuan Ma. Recommendation Method for Learning Path Based on Particle Swarm Optimization [J]. Henan Science, 2013, 31 (12): 2190-2193.
 - [17] Yan C. A Method of Swarm Intelligence-Based Learning Path Recommendation for Online Learning[J]. Journal of Systems & Management, 2011, 20(2): 232-237.
 - [18] Van den Berg B, Van Es R, Tattersall C, et al. Swarm-based Sequencing Recommendations in E-learning[C]. Intelligent Systems Design and Applications, IEEE, 2005: 488-493.
 - [19] Chun Fu Lina, Yu-chuYehb, Yu HsinHungc, Ray I Changa. Data mining for providing a personalized learning path in creativity: An application of decision trees[J]. Computers & Education, 2013, 68: 199–210.

- [20] Basu P, Bhattacharya S, Roy S. Online Recommendation of Learning Path for an E-Learner under Virtual University[M]// Distributed Computing and Internet Technology. Springer Berlin Heidelberg, 2013:126-136.
- [21] Bendahmane M, Falaki B E, Benattou M. Individualized Learning Path Through a Services-Oriented Approach[M]// Europe and MENA Cooperation Advances in Information and Communication Technologies. Springer International Publishing, 2017.
- [22] Salehi M, Kamalabadi I N. Hybrid recommendation approach for learning material based on sequential pattern of the accessed material and the learner's preference tree [J]. Knowledge-Based Systems, 2013, 48:57-69.
- [23] Siemens G. Connectivism: A learning theory for the digital age[J]. International Journal of Instructional Technology & Distance Learning, 2004, 2(s101): 3-10.
- [24] Chu K-K, Lee C-I, Tsai R-S. Ontology technology to assist learners' navigation in the concept map learning system[J]. Expert Systems with Applications, 2011, 38 (9): 11293-11299.
- [25] Colace F, De Santo M. Ontology for E-learning: A Bayesian approach[J]. Education, IEEE Transactions on Education, 2010, 53 (2): 223-233.
- [26] Chen C-M, Lee H-M, Chen Y-H. Personalized e-learning system using item response theory[J]. Computers & Education, 2005, 44 (3): 237-255.
- [27] Juan Yang, Zhixing Huang, Hongtao Liu. Multi-faceted Learning Paths Recommendation via Semantic Linked Network[C]. 2010 Sixth International Conference on Semantics, Knowledge and Grids: 50-57.
- [28] Yunhua Hu, Hang Li, Yunbo Cao, Li Teng, Dmitriy Meyerzon, Qinghua Zheng. Automatic extraction of titles from general documents using machine learning[J]. Information Processing and Management, 2006, 42 (5): 1276-1293.
- [29] Lee JH, Segev A. Knowledge maps for e-learning[J]. Computers & Education, 2012, 59 (2): 353-364.
- [30] Zheng Q, Wu Z, Jiang L, et al. A collaborative knowledge construction system design for massive knowledge resources[C]. Computer Supported Cooperative Work in Design, IEEE, 2009: 137-142.
- [31] Ferrer NF, Alfonso JM. Content management for E-learning[M]. Springer Science & Business Media, 2010.
- [32] Hanewald R. Cultivating Life-Long Learning Skills in Undergraduate Students through the Collaborative Creation of Digital Knowledge Maps[J]. Procedia-Social and Behavioral Sciences, 2012, 69: 847-853.
- [33] Lin J-C, Wu K-C. Finding a fitting learning path in e-learning for juvenile[C]. Advanced Learning Technologies, IEEE, 2007: 449-453.
- [34] Guillaume Durand, Nabil Belacel, François Laplante. Graph theory based model for learning path recommendation[J]. Information Sciences, 2013, 251(4): 10-21.
- [35] O'donnell AM, Dansereau DF, Hall RH. Knowledge maps as scaffolds for cognitive processing[J]. Educational Psychology Review, 2002, 14 (1): 71-86.
- [36] Shaw R-S. A study of learning performance of e-learning materials design with knowledge maps[J]. Computers & Education, 2010, 54 (1): 253-264.
- [37] Wan S, Niu Z. A learner oriented learning recommendation approach based on mixed

- concept mapping and immune algorithm [J]. Knowledge-Based Systems, 2016, 103(C):28-40.
- [38] Tam V, Lam E Y, Fung S T. A new framework of concept clustering and learning path optimization to develop the next-generation e-learning systems[J]. Journal of Computers in Education, 2014, 1(4):335-352.
- [39] Hirate Y, Yamana H. Generalized sequential pattern mining with item intervals[J]. Journal of computers, 2006, 1 (3): 51-60.
- [40] Jiawei Han, Micheline Kamber, Jian Pei. Data Mining: Concepts and Techniques, Third Edition[M]. Morgan Kaufmann, 2011.
- [41] Y Wen, G Xu. Knowledge element linking theory[J]. Journal of the China Society for Scientific & Technical Information, 2003, 22(6): 665–670.
- [42] N Henze, W Nejdl. Logically characterizing adaptive educational hypermedia systems[C]. In International Workshop on Adaptive Hypermedia and Adaptive Web-based Systems, 2003, 10(1): 15–28.
- [43] TME Duffy, DHE Jonassen. Constructivism and the technology of instruction: A Conversation[M]: Routledge, 2013.
- [44] Masek WJ, Paterson MS. A faster algorithm computing string edit distances[J]. Journal of Computer and System Sciences, 1980, 20(1): 18-31.

Acknowledgment

It is a project supported by the National Key Research and Development Program of China under Grant No. 2016YFB1000903; the innovation team of the Ministry of Education IRT13035; the National Science Foundation of China under Grant No. 61402392, 61472315, 61428206, 61532015 and 61532004; The project of China Knowledge Centre for Engineering Science and Technology; The Natural Science Basic Research Plan in Shaanxi Province of China under Grant No.2016JM6027, 2016JM6080; Online Education Research Foundation of MOE Research Center for Online Education under Grant No.2016YB165, 2016YB169.

Appendix A

Relevant properties of knowledge unit of the course Assembly language:

No.	KU	Learning dependency	Duration (second)	Attention degree	Learning centrality
#1	Programming language	#2, #14	1825	2.0771	3.6651
#2	Assembly language	#3	924	1.8661	3.0211
#3	Arrangement of Assembly language	#4	1267	1.7172	2.9878

#4	Preliminary knowledge and binary	#5, #10, #11, #17, #19, #21, #26, #29	1677	0.6489	4.1352
#5	Binary operation	#6	603	1.0963	2.2457
#6	Presentation of number	#7	3690	0.4660	1.4369
#7	Presentation of character	#8	895	1.8010	3.1214
#8	Computer system composition	#9, #11, #12	997	1.7119	2.9754
#9	CPU	#10, #16	1298	1.1591	3.4665
#10	Register	#13, #14, #15, #16, #17, #26, #27, #29, #49, #51, #52	1674	0.9719	4.6880
#11	Memory	#13, #14, #15, #18, #19, #20, #21	1317	0.7323	1.4234
#12	External device	#13, #14, #23, #49	542	1.7928	2.0871
#13	The use of the Debug	#14	660	1.9926	1.0284
#14	Instruction sets and addressing modes	#15, #21	619	2.0063	1.5728
#15	Immediate addressing	#16	719	1.3100	1.6554
#16	Register addressing	#18, #29	262	1.9103	1.7865
#17	Effective address	#18, #21, #55, #56	915	1.4320	2.7512
#18	Memory operand addressing mode 1	#19	2582	0.8207	1.6547
#19	Memory operand addressing mode 2	#20	1040	1.7901	1.9008
#20	Instruction jump addressing	#29	2095	0.5937	1.7327
#21	Data transfer instructions	#22, #24	1802	0.6120	2.0283
#22	Stack and instructions	#23, #40	1251	0.7913	2.4341
#23	I/O instructions	#29, #50	1382	1.0757	2.6987
#24	Arithmetic instructions	#25	3380	0.4953	1.2748
#25	Logic instructions	#26	1657	0.4352	1.7667

#26	String processing instructions	#27	1944	0.5878	1.9750
#27	Control transfer instructions	#28, #34, #38	2698	0.8006	1.4728
#28	Processor control instructions	#29	415	1.1901	2.0867
#29	Pseudo operation 1	#30	2038	1.6835	2.4335
#30	Pseudo operation 2	#31	1593	0.8922	1.8976
#31	Assembly language program format	#32	2413	0.8644	2.0571
#32	Assembly language processing	#33	692	1.6601	1.7657
#33	Debugging method of debug	#34, #38	844	1.4705	1.0972
#34	Structure of cycle program	#35	957	0.9037	2.5348
#35	Cycle programming method 1	#36	90	1.4771	1.4332
#36	Cycle programming method 2	#37	2789	0.5258	1.6754
#37	Multi-cycle programming	#40	1969	0.7383	1.1857
#38	Branch programming	#39	3326	0.3093	2.0896
#39	Method of skip list	#40	893	1.3009	2.3514
#40	Subprogram and stack	#41, #42, #45	1892	0.7162	1.6809
#41	Subprogram definition method	#42	2026	0.4988	1.8245
#42	Parameter passing	#43	2602	0.3912	2.5612
#43	Structure definition and enhanced process definition	#44	638	1.6736	1.1380
#44	Nested and recursive subprogram	#45, #47, #48	1774	0.7469	1.7347
#45	Macro Assembly 1	#46	1569	0.9191	2.0965
#46	Macro Assembly 2	#49	1720	0.7016	3.0992
#47	Conditional assembly	#49	3326	0.2453	2.8774
#48	Advanced language structure	#49	1033	1.0523	1.0651
#49	Data transfer of I/O devices	#50, #51	595	1.8034	1.6785
#50	I/O polling	#53	1297	0.9542	1.9227

#51	I/O interrupt	#52, #53, #54	1949	0.6664	2.0475
#52	Interrupt handlers	#53	1852	0.7927	3.1742
#53	Keyboard I/O	#54	2538	1.4742	2.7865
#54	Display I/O	#55	2370	0.6342	1.6799
#55	Assembler	#56	882	1.5307	1.0428
#56	Linker	#57	1097	1.0023	1.1327
#57	Assembly and C mixed programming		1698	0.9204	4.3257

Appendix B

The Edit Distance algorithm:

Input: learning path P_1, P_2 , their path length len_1, len_2

Output: edit distance between the two paths $EditDistance$

```
1:  $e[][] = \text{newint}[len_1+1][len_2+1];$            { build matrix of edit distance }

2: for  $i=0$  to  $len_1$  do

3:    $e[i][0] = i;$                                { initialize first column of the matrix }

4: end for

5: for  $j=0$  to  $len_2$  do

6:    $e[0][j] = j;$                                { initialize t first row of the matrix }

7: end for

8: for  $i=1$  to  $len_1$  do

9:   for  $j=1$  to  $len_2$  do

10:   $cost = P_1[i-1] == P_2[j-1] ? 0 : 1;$       { judge whether  $P_1[i-1]$  and  $P_2[j-1]$  is the same }

11:   $deletion = e[i-1][j] + 1;$  { calculate edit distance of deletion operation between  $P_1[i]$  and  $P_2[j]$  }

12:   $insertion = e[i][j-1] + 1;$  { calculate edit distance of insertion operation between  $P_1[i]$  and  $P_2[j]$  }

13:   $substitution = e[i-1][j-1] + cost;$       { calculate edit distance of substitution operation between  $P_1[i]$  and  $P_2[j]$  }

14:   $d[i][j] = \min(deletion, insertion, substitution);$  { select shortest edit distance between  $P_1[i]$  and  $P_2[j]$  }

15: end for

16: end for

17:  $EditDistance = e[len_1][len_2];$            { get the edit distance between the two paths }

18: return  $EditDistance;$ 
```
